Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 8063962, 9 pages
http://dx.doi.org/10.1155/2016/8063962

Research Article

Transfer Learning for Collaborative Filtering Using a Psychometrics Model

Haijun Zhang,[1] Bo Zhang,[2] Zhoujun Li,[1] Guicheng Shen,[2] and Liping Tian[2]

[1] *State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China*
[2] *School of Information, Beijing Wuzi University, Beijing 101149, China*

Correspondence should be addressed to Haijun Zhang; hjz_666@126.com

Received 29 November 2015; Revised 9 March 2016; Accepted 20 March 2016

Academic Editor: Matteo Gaeta

In a real e-commerce website, usually only a small number of users will give ratings to the items they purchased, and this can lead to the very sparse user-item rating data. The data sparsity issue will greatly limit the recommendation performance of most recommendation algorithms. However, a user may register accounts in many e-commerce websites. If such users' historical purchasing data on these websites can be integrated, the recommendation performance could be improved. But it is difficult to align the users and items between these websites, and thus how to effectively borrow the users' rating data of one website (source domain) to help improve the recommendation performance of another website (target domain) is very challenging. To this end, this paper extended the traditional one-dimensional psychometrics model to multidimension. The extended model can effectively capture users' multiple interests. Based on this multidimensional psychometrics model, we further propose a novel transfer learning algorithm. It can effectively transfer users' rating preferences from the source domain to the target domain. Experimental results show that the proposed method can significantly improve the recommendation performance.

1. Introduction

A recommendation system aims to find the favorite items (goods) for users. Over the past decade, many recommendation algorithms have been proposed, among which collaborative filtering (CF) has attracted much more attention because of its high recommendation accuracy and wide applicability. These CF algorithms can be grouped into two categories: memory-based CF and model-based CF [1, 2]. Memory-based CF mainly includes user-based CF and item-based CF. User-based CF supposes that the test user will like the items that his or her similar users like, and the estimated rating of the test user for the test item is adjusted by the ratings of his or her similar users. Item-based CF deems that the test user will like the other items similar to the items that he or she previously liked, and the estimated rating of the test user for the test item is calculated by the items this user previously rated. Therefore, the key step in memory-based CF is to calculate the similarity of users or items. Usually, the similarity is calculated directly from the user-item rating matrix. In this matrix, each row denotes a feature vector of a user, while each column denotes a feature vector of an item. The similarity between two vectors depends on the elements at which both vectors have a rating. But usually, only a few users evaluated a few items on websites; thus, the user-item rating matrix is very sparse, and the number of commonly rated elements of the two vectors is very small. This results in an unreliable computed similarity and hence a low recommendation accuracy. Furthermore, with the growth of users and items, the complexity of similarity computation increases in a nonlinear fashion, which restricts its scalability.

To solve these problems, a variety of machine learning and data mining models have been proposed, which have led to the development of model-based CF algorithms. The basic idea of model-based CF algorithms is to reduce the dimension of a rating matrix that will combat the scalability and data sparsity problems. Model-based CF mainly contains clustering methods [3, 4], regression methods [5], graph model-based methods, matrix factorization methods, and neural network methods. Graph model-based methods include Bayesian network [6, 7], PLSA (Probabilistic Latent Semantic Analysis) [8, 9], and LDA (Latent Dirichlet Allocation) [10]. Representative Matrix factorization methods include SVD (Singular Value Decomposition) [11–15], PCA (Principal Component Analysis) [16], and NMF (Nonnegative Matrix Factorization) [17, 18]. Neural network methods include RBM (Restrained Boltzmann Machine) [19]. Generally, model-based CF has better performance than memory-based CF; however, the recommendation produced by memory-based CF has better interpretability and makes convincing users easier. However, these methods only utilize the rating data coming from a single website and ignore exploiting the ratings from other websites.

In order to exploit big data gathered from different websites to cope with data sparsity problems, recently, the transfer learning methods [22–29] were proposed. They utilize the data from the source domain to improve the prediction accuracy of the target domain. However, transfer learning can potentially hinder performance if there is a significant difference between the source domain and the target domain [30]. For successful knowledge transfer, there are two critical problems that must be resolved. First, the two domains share the same feature space (in recommendation system, it means that the two domains have aligned users or items); however, their distributions are significantly different. Second, the two domains do not share the same feature space (in recommendation system, it means that the two domains have no aligned users or items). To solve the first problem, some methods [24–26, 31] were proposed to supervise the knowledge transfer. The second problem is very difficult, because a common latent space is needed in which the two domains are similar, and the common factors learned from the source domain can be used to constrain the algorithm to achieve better performance in the target domain. A representative method to solve this tough problem is CBT (CodeBook Transfer), proposed in [23]. The algorithm CBT exploits nonnegative matrix trifactorization to transfer information from the source domain to the target domain, which can be expressed by

$$\begin{aligned} \min_{U_s\geq 0, B_s\geq 0, V_s\geq 0} \quad & \left\| R_s - U_s B_s {V_s}^T \right\|_F^2 \\ \text{s.t.} \quad & {U_s}^T U_s = I, \\ & {V_s}^T V_s = I \end{aligned} \tag{1}$$

$$\begin{aligned} \min_{U_t\geq 0, B_t\geq 0, V_t\geq 0} \quad & \left\| R_t - U_t B_t {V_t}^T \right\|_F^2 + \lambda \left\| B_s - B_t \right\|_F^2 \\ \text{s.t.} \quad & {U_t}^T U_t = I, \\ & {V_t}^T V_t = I, \end{aligned} \tag{2}$$

where R_s denotes the rating matrix in the source domain, R_t denotes the rating matrix in the target domain, and $\lambda > 0$ is a tradeoff parameter to balance the target and source domain data. U_s, B_s, and V_s are nonnegative matrices. B_s matrix is learned from the source domain according to (1), and B_s is then fixed to learn the factor matrices of the target domain by (2). This method expects to find a common latent space (here, it is represented by the middle matrix) in which the information obtained from the source domain data (i.e., matrix B_s) can be used to improve the algorithm's performance in the target domain. However, there are several implementation methods of nonnegative matrix trifactorization. Some constraints are usually needed in these methods to guarantee that the columns of the left matrix and right matrix are orthogonal [17, 24, 25, 32]. Note that too strict or too loose constraints would damage the performance of the nonnegative matrix trifactorization based transfer learning method.

Compared with our previous work [33], the differences between them and the new contributions of this paper are as follows. (1) In our previous work, a multidimensional psychometrics-based collaborative filtering algorithm was proposed, which is presented in Section 2.1. (2) In this paper, we extend the algorithm and propose a novel transfer learning algorithm in Section 2.2. This new proposed method can effectively transfer knowledge from source domain to target domain even when the two domains have no aligned users or items. In Section 3.2, MovieLens dataset and BookCrossing dataset are used to evaluate the two algorithms to validate the effectiveness of knowledge transfer. Comprehensive experiments are conducted in Section 3.3, and the results show that our novel transfer learning algorithm outperforms representative state-of-the-art algorithms including nonnegative matrix trifactorization based transfer learning method [23].

2. Materials and Methods

In this section, a multidimensional psychometrics model is proposed. Compared with one-dimensional psychometrics model, it can model user's multilatent interests. Furthermore, a novel transfer learning method based on this multidimensional psychometrics model is proposed, which can mine rating preferences parameters and transfer them from source domain to target domain to improve the recommendation accuracy further.

2.1. Collaborative Filtering Using a Multidimensional Psychometrics Model. Hu et al. [20, 21] first introduced the psychometrics model into CF and used the software Winsteps [34] to estimate the parameters in the model. It can be expressed as

$$\log \frac{P_{uik}}{P_{ui(k-1)}} = B_u - D_i - F_k \quad k = 2, 3, \ldots, K, \tag{3}$$

$$F_1 = 0, \tag{4}$$

where B_u and D_i are scalar quantities, which denote the interest of user u and the drawback of item i. Higher B_u means that user u has a great interest in the items, and he or she will assign a higher score to the items. Smaller D_i indicates that item i is of good quality and will receive higher scores from users. The parameter P_{uik} denotes the probability of user u assigning score k to item i, and F_k is the ordered threshold, which denotes the difficulty for users to give score k relative to giving score $k-1$ for an item. Each item has K-level scores, that is, $1, 2, \ldots, K$, and parameters B_u, D_i, and $F_2 \sim F_K$ need to be estimated.

In (3), B_u and D_i are scalar quantities and denote the user's interest and the item's quality, respectively. The model expressed by (3) assumes that user's interest and item's quality are one-dimensional. However, it is more feasible to represent the user's interest and the item's quality with vectors because users usually have varied interests and gain satisfaction from item in a number of different aspects [33]. For example, a user can have an interest in science fiction films with a value of 2, an interest in action movies with a value of 8, an interest in romantic films with a value of 1, an interest in movie scenes with a value of 3, and an interest in actors with a value of 5. For a film that is 30% science fiction and 20% action, the scene quality could earn a value of 2 while the actors' performance earns a value of 3. A user's rating for a movie is determined from a comprehensive assessment of the satisfaction that the movie brings to him or her in the various dimensions. Based on this idea, the psychometrics model is extended to multidimension, which is expressed as

$$\log \frac{P_{uikl}}{P_{ui(k-1)l}} = B_{ul} - D_{il} - F_k \qquad k = 2, 3, \ldots, K;\ l = 1, 2, \ldots, L, \qquad F_1 = 0, \tag{5}$$

where P_{uikl} is the probability of user u assigning rating score k to item i in dimension l and B_{ul} is the interest value of user u in dimension l. $B_u = (B_{u1}, B_{u2}, \ldots, B_{uL})^T$ denotes the interest vector of user u. A high value of B_{ul} indicates that user u has a strong interest in dimension l, which implies that if a film can satisfy a user's interest in this area, then he or she will assign a high latent rating value to this film in this dimension. D_{il} is the quality value of item i in dimension l; therefore, $D_i = (D_{i1}, D_{i2}, \ldots, D_{iL})^T$ can denote the quality vector of item i. Higher D_{il} indicates that item i has *less* content related to dimension l and is less probable of earning a high rating in dimension l, and F_k is the ordered threshold that measures the difficulty for all users to assign k points relative to assigning $k-1$ points to an item. Each item has a K-level rating. From the MovieLens (http://www.grouplens.org/) dataset, K is set to 5. L is the number of dimensions, which denotes the number of latent interests or demands each user has, and it is a metaparameter, which should be set in advance. If L is set to 1, this model will degenerate into the model represented by (3). Parameters B_{ul}, D_{il}, and F_k should be estimated using the training set. To simplify, the notation ψ_k is introduced, and (6)–(8) can be deduced from (5). Hence,

$$\psi_k = -\sum_{j=1}^{k} F_j, \tag{6}$$

$$\psi_1 = 0, \tag{7}$$

$$P_{uikl} = \frac{e^{\psi_k + (k-1)(B_{ul} - D_{il})}}{\sum_{j=1}^{K} e^{\psi_j + (j-1)(B_{ul} - D_{il})}} \qquad k = 1, 2, \ldots, K;\ l = 1, 2, \ldots, L. \tag{8}$$

Equation (8) shows that the rating of user u for item i in dimension l is of a multinomial distribution with K categories determined by B_{ul}, D_{il}, and ψ_k. After P_{uikl} $(k = 1, 2, \ldots, K)$ is obtained, there are two methods to estimate the latent rating of user u for item i in dimension l (denoted by $\hat{r}_{uil}$), and they are as follows:

(1) The rating score corresponding to the biggest probability:

$$\hat{r}_{uil} = \arg\max_k P_{uikl}. \tag{9}$$

(2) The expectation given by

$$\hat{r}_{uil} = \sum_{k=1}^{K} k \cdot P_{uikl}. \tag{10}$$

In this paper, for simplicity, the second method is used to estimate the latent rating of user u for item i in dimension l because the function of $\hat{r}_{uil}$ described by (10) is continuous while the function of (9) is not continuous. After $\hat{r}_{uil}$ $(l = 1, 2, \ldots, L)$ is obtained, there are three methods to estimate the rating value of user u for item i, and they are formulated by (11)–(13). In this paper, for simplicity, (13) is used to estimate the rating value of user u for item i. The three methods are as follows:

(1) The maximum value of $\hat{r}_{uil}$ $(l = 1, 2, \ldots, L)$ as follows:

$$\hat{r}_{ui} = \max_l \hat{r}_{uil}. \tag{11}$$

(2) The minimum value of $\hat{r}_{uil}$ $(l = 1, 2, \ldots, L)$ as follows:

$$\hat{r}_{ui} = \min_l \hat{r}_{uil}. \tag{12}$$

(3) The weighted summation of $\hat{r}_{uil}$ $(l = 1, 2, \ldots, L)$ as follows:

$$\hat{r}_{ui} = \frac{\sum_{l=1}^{L} w_{ul} \cdot \hat{r}_{uil}}{\sum_{l=1}^{L} w_{ul}}, \tag{13}$$

where $w_{ul} \geq 0$ and denotes the weight of user u in dimension l. It also represents the importance of the lth interest or demand for user u; therefore, the weight vector of user u can be determined by $w_u = \{w_{u1}, w_{u2}, \ldots, w_{uL}\}^T$, and it will be estimated using the training set.

In order to learn the parameters, the loss function is designed as

$$f = \sum_{(u,i)\in \mathrm{Tr}} \left(r_{ui} - \hat{r}_{ui}\right)^2 + \lambda \left(\sum_{u,l} {w_{ul}}^2 + \sum_{u,l} {B_{ul}}^2 + \sum_{i,l} {D_{il}}^2 + \sum_{j} {\psi_j}^2 \right), \tag{14}$$

where Tr denotes the training set. The first term on the right-hand side of the equation is the sum of the square errors, and the other four terms are the regularization terms that are applied to the parameters to avoid overfitting. The regularization constant λ is a parameter that requires tuning.

It is easy to find that the partial derivative of the loss function is an elementary function; therefore, the partial derivative of the loss function is continuous, and the loss function is differentiable. Thus, the stochastic gradient descent method [14] can be used to learn the parameters. Convergence of the stochastic gradient descent method has been empirically verified in many previous works [14, 15, 35], given the appropriate learning ratio parameters. Each rating r_{ui} in the training set Tr is used to update the parameters according to (17)–(20). Parameter ψ_1 should not be updated and is fixed as 0 according to (7). Learning ratio parameters γ_1, γ_2, and γ_3 are parameters that should be tuned. For easy representation, notations e_{ui} and ψ_{juil} are introduced by (15) and (16). Hence, we have the following:

$$e_{ui} = r_{ui} - \widehat{r}_{ui}, \tag{15}$$

$$\psi_{juil} = e^{\psi_j + (j-1)(B_{ul} - D_{il})} \quad \text{for } j = 1, 2, \ldots, K;\ l = 1, 2, \ldots, L, \tag{16}$$

$$\begin{aligned} w_{ul} \longleftarrow \max\Biggl(0, w_{ul} \\ + \gamma_1 \left(e_{ui} \left(\frac{\widehat{r}_{uil} \sum_{l=1}^{L} w_{ul} - \sum_{l=1}^{L} \left(w_{ul} \widehat{r}_{uil} \right)}{\left(\sum_{l=1}^{L} w_{ul} \right)^2} \right) - \lambda w_{ul} \right) \Biggr), \end{aligned} \tag{17}$$

$$\begin{aligned} B_{ul} \longleftarrow B_{ul} + \gamma_2 \Biggl(\frac{e_{ui}}{\sum_{l=1}^{L} w_{ul}} w_{ul} \Biggl(\sum_{k=1}^{K} k \\ \cdot \frac{(k-1)\psi_{kuil} \left(\sum_{m=1}^{K} \psi_{muil} \right) - \psi_{kuil} \left(\sum_{m=1}^{K} (m-1) \psi_{muil} \right)}{\left(\sum_{m=1}^{K} \psi_{muil} \right)^2} \Biggr) \\ - \lambda B_{ul} \Biggr), \end{aligned} \tag{18}$$

$$\begin{aligned} D_{il} \longleftarrow D_{il} + \gamma_2 \Biggl(\frac{e_{ui}}{\sum_{l=1}^{L} w_{ul}} w_{ul} \Biggl(\sum_{k=1}^{K} k \\ \cdot \frac{(1-k)\psi_{kuil} \left(\sum_{m=1}^{K} \psi_{muil} \right) - \psi_{kuil} \left(\sum_{m=1}^{K} (1-m) \psi_{muil} \right)}{\left(\sum_{m=1}^{K} \psi_{muil} \right)^2} \Biggr) \\ - \lambda D_{il} \Biggr), \end{aligned} \tag{19}$$

$$\begin{aligned} \psi_j \longleftarrow \psi_j + \gamma_3 \Biggl(\frac{e_{ui}}{\sum_{l=1}^{L} w_{ul}} \Biggl(\sum_{l=1}^{L} w_{ul} \\ \cdot \frac{j\psi_{juil} \left(\sum_{m=1}^{K} \psi_{muil} \right) - \left(\sum_{m=1}^{K} m\psi_{muil} \right) \psi_{juil}}{\left(\sum_{m=1}^{K} \psi_{muil} \right)^2} \Biggr) - \lambda \psi_j \Biggr) \\ \text{for } j = 2, \ldots, K. \end{aligned} \tag{20}$$

The pseudo codes of the collaborative filtering using multidimensional psychometrics model (CFMPM) are shown in Algorithm 1. After the parameters have been estimated by the algorithm, the ratings in the test set can be predicted by (13).

2.2. Transfer Learning Based on Multidimensional Psychometrics Model.

In this section, we will represent the notion that, in the multidimensional psychometrics model, the rating preferences parameters of all the datasets are similar, and these parameters may be the common knowledge that could be shared from source domain to target domain. Based on this idea, a novel transfer learning method is proposed.

Equations (21)–(25) can be derived from (8), (10), and (13), where ψ_k is defined by (6) and (7), and the notation "≅" means that the two sides of the equation have the same sign. P_{uik} denotes the probability for user u to give score k to item i, and P_k denotes the probability of getting score k for an item. The notation "∝" means that the two sides of the equation have the same increase or decrease trend, and the notation $|r_{ui} = k|$ denotes the number of ratings k in the training set. If the entry (u, i) is observed in the training set, $y_{ui} = 1$, or not, $y_{ui} = 0$,

$$\begin{aligned} \frac{\partial P_{uikl}}{\partial (\psi_k)} \cong \left(\sum_{j=1}^{K} e^{\psi_j + (j-1)(B_{ul} - D_{il})} \right) - e^{\psi_k + (k-1)(B_{ul} - D_{il})} \\ > 0, \end{aligned} \tag{21}$$

$$P_{uik} = \frac{\sum_{l=1}^{L} w_{ul} P_{uikl}}{\sum_{l=1}^{L} w_{ul}}, \tag{22}$$

$$\frac{\partial P_{uik}}{\partial (\psi_k)} > 0, \tag{23}$$

$$P_k \propto \left| r_{ui} = k \right| \propto \sum_{u} \sum_{i} y_{ui} P_{uik}, \tag{24}$$

$$\frac{\partial P_k}{\partial (\psi_k)} > 0. \tag{25}$$

From (25), we can see that the sign of the partial derivative of P_k with respect to ψ_k is constant and positive, which means that when ψ_k increases, users have a higher probability (i.e., P_k) of assigning score k to item i and vice versa. It also means that a higher value of P_k will result in a higher value of ψ_k, and a lower value of P_k will result in a lower value of ψ_k. In reality, users rarely assign a bad evaluation. Users usually give a score of 4 or 5 (in a 1–5-scaled rating system) to an item because if a user does not like the item, the user will not buy it at all. The proportion of each rating score in three different datasets is shown in Figure 1, which shows that the rating score distributions in the three datasets are similar. It means that the users in the different domains have similar rating preferences; that is, the threshold ψ_k is usually similar in different domains. Based on this idea, if we learn parameter ψ_k from the source domain and transfer it to the target domain, the prediction accuracy in the target domain will possibly be improved. Because of the similarity of parameter

```
Input: Training set Tr
Output: $B_u$, $D_i$, $w_u$, and $\psi_k$ ($k = 2, 3, \ldots, K$)
Initialize parameters $B_u$, $D_i$, $w_u$, $\psi_k$, $\gamma_1$, $\gamma_2$, $\gamma_3$, and $\lambda$.
For each iteration
    Randomly permutate the data in Tr.
    For each $r_{ui}$ in Tr
        Compute $e_{ui}$ by (15);
        Compute $\psi_{juil}$ ($j = 1, 2, \ldots, K; l = 1, 2, \ldots, L$) by (16);
        Update the parameters according to (17)–(20).
    End for
    Decrease $\gamma_1$, $\gamma_2$, $\gamma_3$; they are multiplied by 0.9.
End for
```

Algorithm 1: Collaborative filtering using multidimensional psychometrics model (CFMPM).

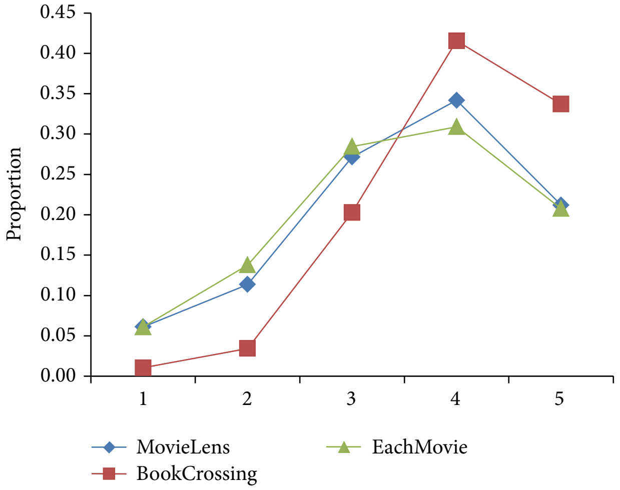


Figure 1: The proportion of each rating score on the three datasets (e.g., about 5 percent of all the ratings in MovieLens and EachMovie are 1).

ψ_k across different domains, the parameter ψ_k learned from the source domain contains the information about the target domain. Meanwhile, transferring ψ_k into the target domain will increase the model's generalization capability according to Occam's Razor, as it reduces the number of parameters to be learned. Therefore, a new transfer learning method is designed as shown in Algorithm 2, which is called TLMPM (transfer learning based on multidimensional psychometrics model).

3. Results

Two experiments are designed to evaluate the performance of the algorithm TLMPM. The first one is to estimate the sensitivity of parameter ψ_k and to compare algorithms TLMPM and CFMPM with the collaborative filtering algorithms proposed in [20, 21], which are based on one-dimensional psychometrics. The other experiment is used to compare TLMPM with other representative state-of-the-art algorithms including nonnegative matrix trifactorization based transfer learning method CBT, which is reported in [23].

3.1. Datasets and Evaluation Metrics.

3.1. Datasets and Evaluation Metrics. We use the following datasets in our empirical tests.

EachMovie (http://www.cs.cmu.edu/~lebanon/IR-lab.htm) (source domain) is a movie rating dataset that contains 2.8 million ratings (scales 0–5) rated by 72,916 users for 1,628 movies. Because user's implicit feedback is denoted by score 0 in the EachMovie dataset and this implicit feedback has many noises, we removed the score 0. Similar to Li et al. [23], we extract a submatrix to simulate the source domain by selecting 500 users and 500 movies with the most ratings (rating ratio 54.67%).

MovieLens (http://www.grouplens.org/) (target domain) is a movie rating dataset that contains 100,000 rating scores (1–5 scales) rated by 943 users for 1682 items, where each user has more than 20 ratings.

BookCrossing (http://www.informatik.uni-freiburg.de/~cziegler/BX/) (target domain) contains 77,805 users, 433,671 ratings (scales 1–10), and approximately 185,973 books. We normalize the rating scales from 1 to 5.

We use Mean Absolute Error (MAE) as the evaluation metric. It is used to average the absolute deviation between the predicted values and the true values:

$$\mathrm{MAE} = \frac{\sum_{(u,i)\in \mathrm{Te}} \left| r_{ui} - \hat{r}_{ui} \right|}{|\mathrm{Te}|}, \tag{26}$$

where |Te| is the number of tested ratings in test set Te. The lower the MAE is, the better the performance is.

3.2. Performance Comparison between Psychometrics-Based CF Algorithms. In this section, we will evaluate the proposed algorithms using the MovieLens dataset and the BookCrossing dataset. Here, we named the one-dimensional psychometrics model-based algorithms proposed in [20, 21] CFOPM_1 and CFOPM_2, respectively. To estimate the necessity of the parameter ψ_k in our algorithm CFMPM, a method called CFMPM_L is designed which differs from CFMPM only in that its parameter ψ_k is set to 0, and it will always remain unchanged. Fivefold cross-validation

```
Input: The data of the source domain denoted by $\mathrm{Tr}_{\mathrm{src}}$ and the training data set of the target domain denoted by $\mathrm{Tr}_{\mathrm{tgt}}$.
Output: Parameters $B_u$, $D_i$, $w_u$, and $\psi_k$ $(k = 2, 3, \ldots, K)$ of the target domain.
(1) Learn the parameters of the source domain by algorithm CFMPM using $\mathrm{Tr}_{\mathrm{src}}$, and get parameters $\psi_k$ $(k = 2, 3, \ldots, K)$ of the
    source domain.
(2) Set parameters $\psi_k$ $(k = 2, 3, \ldots, K)$ of the target domain equal to the parameters $\psi_k$ $(k = 2, 3, \ldots, K)$ of the source domain,
    respectively.
(3) Initialize parameters $B_u$, $D_i$, $w_u$, $\gamma_1$, $\gamma_2$ and $\lambda$ of the target domain.
(4) For each iteration
(5)     Randomly permutate the data in $\mathrm{Tr}_{\mathrm{tgt}}$.
(6)     For each $r_{ui}$ in $\mathrm{Tr}_{\mathrm{tgt}}$.
(7)         Compute $e_{ui}$ by (15);
(8)         Compute $\psi_{juil}$ $(j = 1, 2, \ldots, K; l = 1, 2, \ldots, L)$ by (16);
(9)         Update the parameters according to (17)–(19).
(10)    End for
(11)    Decrease $\gamma_1$, $\gamma_2$; they are multiplied by 0.9.
(12) End for
```

Algorithm 2: Transfer learning based on multidimensional psychometrics model (TLMPM).

Table 1: Comparison on MAE with the algorithms reported in [20, 21].

	TLMPM	CFMPM_L	CFMPM	CFOPM_1	CFOPM_2
MovieLens	**0.7215**	0.7329	0.7249	0.80	0.7250
BookCrossing	**0.6112**	0.6384	0.6147	0.6556	0.6871

was performed for each dataset, and, for every test set, the algorithms were run independently 5 times, and the average MAE over 25 times is reported. In this experiment, the parameters in TLMPM, CFMPM_L, and CFMPM are set as follows: the initial values of B_{ul} and D_{il} are sampled from the uniform distribution on $[0, 0.5]$; the initial value of w_{ul} is set to 1; and the initial value of ψ_k $(k = 1, 2, \ldots, K)$ is set to 0; the values in $\{0.001, 0.01, 0.1\}$ are used to tune the parameter λ; and the values in $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5\}$ are used to tune the parameters γ_1, γ_2, and γ_3; the values in $\{5, 10, 15, 20, 25\}$ are used to tune L, the dimension of latent interests. To evaluate algorithm TLMPM, firstly, algorithm CFMPM is used to learn parameters ψ_k $(k = 2, 3, \ldots, K)$ from EachMovie dataset as described in the previous section, and then the learned parameters ψ_k $(k = 2, 3, \ldots, K)$ are copied to the target domain and held constant to learn the other parameters. The results are shown in Table 1. In Table 1, the results of CFOPM_1 and CFOPM_2 for the MovieLens dataset are extracted directly from Hu's papers [20, 21]. The MAE values for these two algorithms on BookCrossing dataset are our experimental results, and our experimental settings are according to the papers of Hu et al. [20, 21].

From Table 1, it is shown that, of the two datasets, algorithm CFMPM outperforms CFOPM_1. Although the MAE of CFMPM is not significantly lower than that of CFOPM_2 on MovieLens, CFMPM has a significantly lower MAE than CFOPM_2 has on BookCrossing dataset. The time complexity of CFMPM is $O(TNL^2K^2)$, while the time complexity of CFOPM_2 is $O(U^2V)$. Here, T is the iterate times, N is the number of ratings in training set, L is the dimensionality of user's interest, K is the rating levels, U is the number of users, and V is the number of items. In our experiments, L is smaller than 25 and T is smaller than 30. Therefore, on the MovieLens dataset, the time complexity of CFMPM is $O(4.69E + 10)$ and it is $O(1.5E + 09)$ for CFOPM_2. On the BookCrossing dataset, the time complexities for CFMPM and CFOPM_2 are $O(2.03E + 11)$ and $O(1.13E + 15)$, respectively. Thus, if the rating matrix is extremely sparse, for example, BookCrossing, CFMPM will have a lower computational time than CFOPM_2. This means that extending the psychometrics model from one dimension to multidimension is effective. Meanwhile, algorithm CFMPM outperforms CFMPM_L, which verifies that parameters ψ_k $(k = 1, \ldots, 5)$ are essential in this model, although the number of these parameters is only 5. Furthermore, the algorithm TLMPM outperforms CFMPM, which means that the transfer is positive even though only four parameters (i.e., ψ_2, ψ_3, ψ_4, and ψ_5) were transferred from the source domain to the target domain.

3.3. Performance Comparison between TLMPM and Other State-of-the-Art Algorithms.

The compared algorithms are Codebook Transfer (CBT) [23] and the other baseline algorithms used in [23], which are Scalable Cluster-Based Smoothing (CBS) [36] and user-based CF based on Pearson correlation coefficients (PCC) similarity [37]. Algorithm CBT [23] is a representative transfer learning method based on nonnegative matrix trifactorization. Except for TLMPM and CBT, the other algorithms only use the data in the target domain to predict the ratings in the test set.

To consistently compare algorithm TLMPM with the other algorithms described above, as in [23], we also extract a subset of the MovieLens dataset, which contains 1000 items and 500 users, where each user has more than 40 ratings (rating ratio 10.35%). For each run of algorithm TLMPM, the

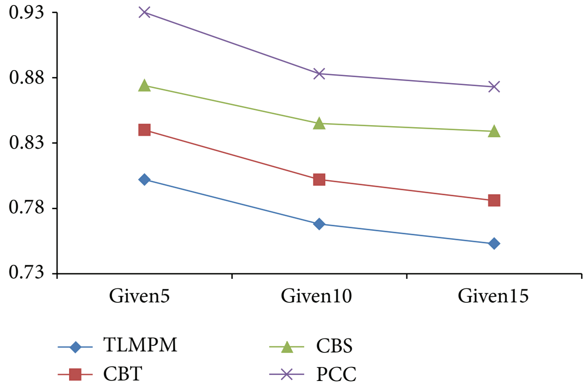


FIGURE 2: MAE on the dataset ML_100.

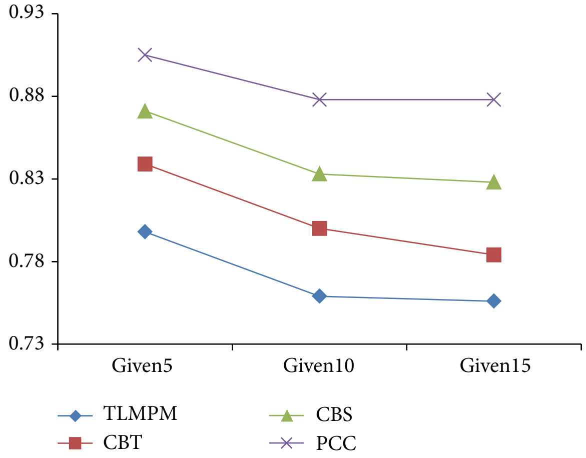


FIGURE 3: MAE on the dataset ML_200.

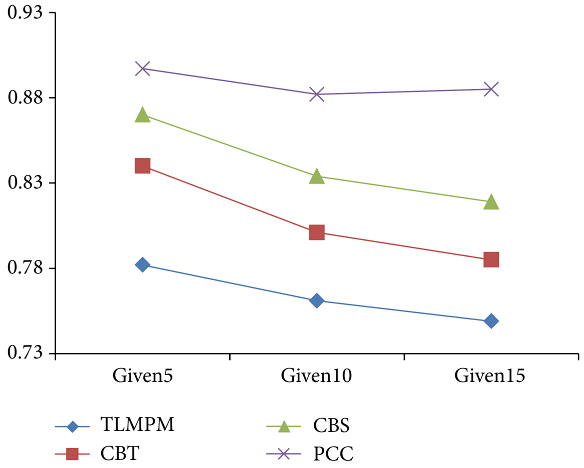


FIGURE 4: MAE on the dataset ML_300.

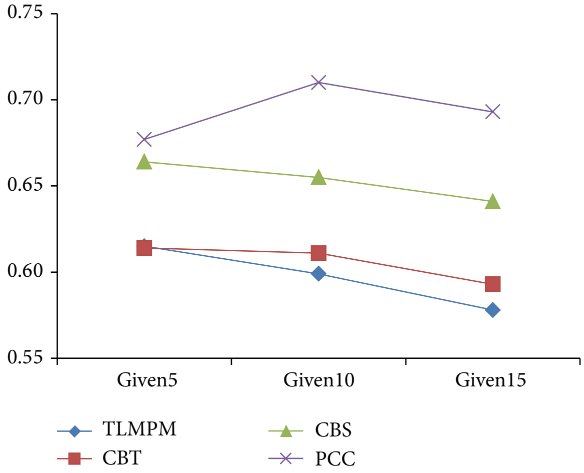


FIGURE 5: MAE on the dataset BX_100.

training set and the test set are generated as follows. First, 500 users were randomly sorted. Second, the first 100, 200, or 300 users were selected as training sets and denoted as ML_100, ML_200, and ML_300, respectively, and the test set contains the last 200 users. Additionally, in order to evaluate the performance of the algorithm at different data sparsity levels, as [23] did, the available ratings of each test user are also split into an observed set and a held-out set. The observed ratings are used to predict the held-out ratings. We randomly selected 5, 10, and 15 items rated by each test user to constitute the observed set, and they are named Given5, Given10, and Given15, respectively, while the other items rated by the test user constitute the held-out set. The BookCrossing dataset is also preprocessed using the same method described above and then produces three training datasets denoted by BX_100, BX_200, and BX_300, respectively.

In this experiment, the parameters in algorithm TLMPM are initialized in the same way as described in the previous section. The comparison results are shown in Figures 2–7, and the reported results of algorithm TLMPM are the average MAE over 10 independent run times.

From Figures 2–7, we can see that the algorithm TLMPM almost outperforms all the other algorithms at each sparsity level on all datasets, which verifies that transferring rating preferences from the source domain to the target domain is effective.

4. Discussion

This paper extended the psychometric model from one dimension to multidimension to model user's multi-interests and then proposed algorithm CFMPM, which outperforms the algorithm based on the one-dimensional psychometric model. After that, a novel transfer learning algorithm based on CFMPM was developed. In this transfer algorithm, the psychometric model is used to learn user's rating preferences from the source domain and then transfer them to the target domain. This transfer algorithm does not require two domains having overlapped users or items. Experimental results show that it is a positive transfer and that it outperforms other representative state-of-the-art algorithms such as CBT, the transfer algorithm based on nonnegative matrix trifactorization. But, in a 5-scaled rating system, the number of transferred parameters is only 4, which limits the algorithm's effectiveness. In future work, we intend to

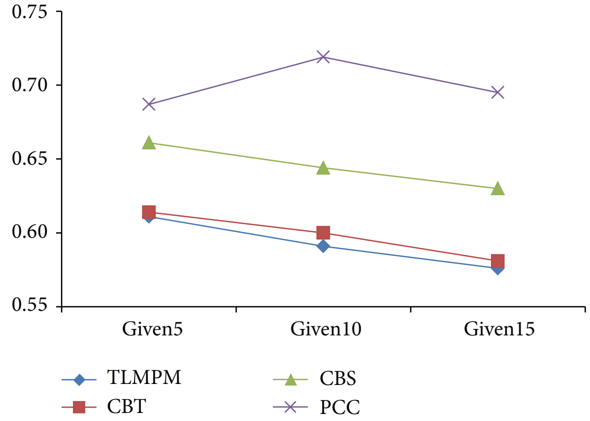


FIGURE 6: MAE on the dataset BX_200.

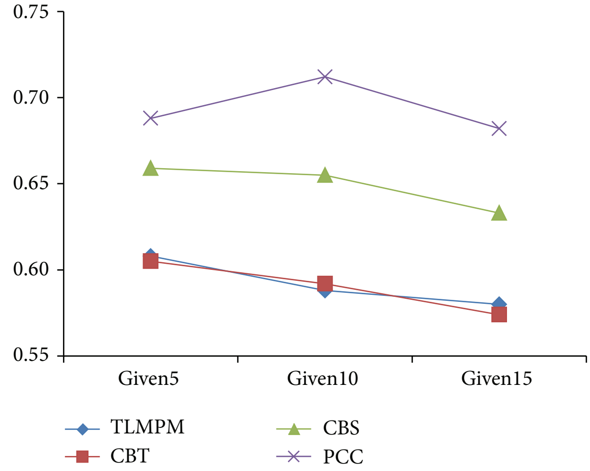


FIGURE 7: MAE on the dataset BX_300.

study how to break this limitation and how to speed up the algorithm.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

The authors appreciate Yan Chen, Senzhang Wang, Yang Yang, and Xiaoming Zhang for their constructive suggestions for polishing this paper. This work is supported by the National Natural Science Foundation of China (nos. 61370126, 71540028, and 61571052), the Fund of Beijing Social Science (no. 14JGC103), the Statistics Research Project of National Bureau (no. 2013LY055), the Fund of Beijing Key Laboratory (no. BZ0211), and the High-Level Research Projects' Cultivation Fund of Beijing Wuzi University (no. GJB20141002).

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[2] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The Adaptive Web*, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., vol. 4321 of *Lecture Notes in Computer Science*, pp. 291–324, Springer, Berlin, Germany, 2007.

[3] M. O'Connor and J. Herlocker, "Clustering items for collaborative filtering," in *Proceedings of the ACM SIGIR Workshop on Recommender Systems (SIGIR '99)*, Berkeley, Calif, USA, August 1999.

[4] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Recommender systems for large-scale e-commerce: scalable neighborhood formation using clustering," in *Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT '02)*, 2002.

[5] S. Vucetic and Z. Obradovic, "Collaborative filtering using a regression-based approach," *Knowledge and Information Systems*, vol. 7, no. 1, pp. 1–22, 2005.

[6] X. Su and T. M. Khoshgoftaar, "Collaborative filtering for multi-class data using belief nets algorithms," in *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '06)*, pp. 497–504, IEEE, Arlington, Va, USA, October 2006.

[7] K. Miyahara and M. J. Pazzani, "Collaborative filtering with the simple Bayesian classifier," in *PRICAI 2000 Topics in Artificial Intelligence: 6th Pacific Rim International Conference on Artificial Intelligence Melbourne, Australia, August 28–September 1, 2000 Proceedings*, vol. 1886 of *Lecture Notes in Computer Science*, pp. 679–689, Springer, Berlin, Germany, 2000.

[8] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 89–115, 2004.

[9] T. Hofmann and J. Puzicha, "Latent class models for collaborative filtering," in *Proceedings of the International Joint Conference on Artificial Intelligence*, vol. 16, pp. 688–693, 1999.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.

[11] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system—a case study," in *Proceedings of the ACM KDD Workshop on Web Mining for e-Commerce-Challenges and Opportunities*, Boston, Mass, USA, August 2000.

[12] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Incremental singular value decomposition algorithms for highly scalable recommender systems," in *Proceedings of the 5th International Conference on Computer and Information Science*, 2002.

[13] S. Funk, "Netflix update: try this at home," Tech. Rep., 2006, http://www.sifter.org/~simon/journal/20061211.html.

[14] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 426–434, Las Vegas, Nev, USA, August 2008.

[15] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[16] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: a constant time collaborative filtering algorithm," *Information Retrieval*, vol. 4, no. 2, pp. 133–151, 2001.

[17] G. Chen, F. Wang, and C. Zhang, "Collaborative filtering using orthogonal nonnegative matrix tri-factorization," *Information Processing & Management*, vol. 45, no. 3, pp. 368–379, 2009.

[18] T. C. Zhou, H. Ma, I. King et al., "Tagrec: leveraging tagging wisdom for recommendation," in *Proceedings of the International Conference on Computational Science and Engineering (CSE '09)*, vol. 4, pp. 194–199, Vancouver, Canada, August 2009.

[19] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pp. 791–798, Corvallis, Ore, USA, June 2007.

[20] B. Hu, Z. Li, and J. Wang, "User's latent interest-based collaborative filtering," in *Proceedings of 32nd European Conference on Information Retrieval: 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28–31, 2010.Proceedings*, vol. 5993 of *Lecture Notes in Computer Science*, pp. 619–622, Springer, Berlin, Germany, 2010.

[21] B. Hu, Z. Li, W. Chao, X. Hu, and J. Wang, "User preference representation based on psychometric models," in *Proceedings of the 22nd Australia Database Conference (ADC '11)*, pp. 59–66, Perth, Australia, 2011.

[22] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[23] B. Li, Q. Yang, and X. Xue, "Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction," in *Proceedings of the 21st International Joint Conference on Artifical Intelligence*, pp. 2052–2057, Pasadena, Calif, USA, July 2009.

[24] W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang, "Transfer learning in collaborative filtering for sparsity reduction," in *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.

[25] W. Pan, N. N. Liu, E. W. Xiang, and Q. Yang, "Transfer learning to predict missing ratings via heterogeneous user feedbacks," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 2318–2323, Barcelona, Spain, July 2011.

[26] W. Pan, E. W. Xiang, and Q. Yang, "Transfer learning in collaborative filtering with uncertain ratings," in *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pp. 662–668, Toronto, Canada, July 2012.

[27] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: a survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1019–1034, 2015.

[28] Z. Deng, K.-S. Choi, Y. Jiang, and S. Wang, "Generalized hidden-mapping ridge regression, knowledge-leveraged inductive transfer learning for neural networks, fuzzy systems and kernel methods," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2585–2599, 2014.

[29] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: a survey," *Knowledge-Based Systems*, vol. 80, pp. 14–23, 2015.

[30] M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich, "To transfer or not to transfer," in *Proceedings of the Workshop on Inductive Transfer: 10 Years Later (NIPS '05)*, vol. 2, p. 7, December 2005.

[31] X. Shi, W. Fan, and J. Ren, "Actively transfer domain knowledge," in *Machine Learning and Knowledge Discovery in Databases*, pp. 342–357, Springer, Berlin, Germany, 2008.

[32] C. Ding, T. Li, W. Peng et al., "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 126–135, ACM, Philadelphia, Pa, USA, August 2006.

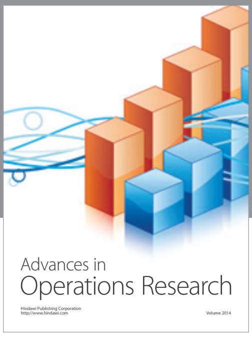
[33] H. Zhang, X. Zhang, Z. Li, and C. Liu, "Collaborative filtering using multidimensional psychometrics model," in *Web-Age Information Management*, J. Wang, H. Xiong, Y. Ishikawa, J. Xu, and J. Zhou, Eds., vol. 7923 of *Lecture Notes in Computer Science*, pp. 691–697, 2013.

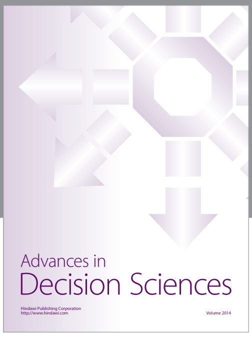
[34] J. M. Linacre, *WINSTEPS: Rasch Measurement Computer Program*, Winsteps, Chicago, Ill, USA, 2007.

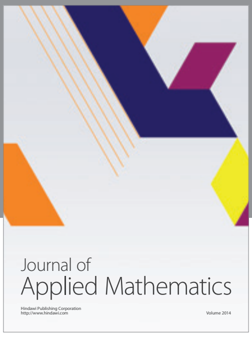
[35] Y. Koren, "Collaborative filtering with temporal dynamics," *Communications of the ACM*, vol. 53, no. 4, pp. 89–97, 2010.

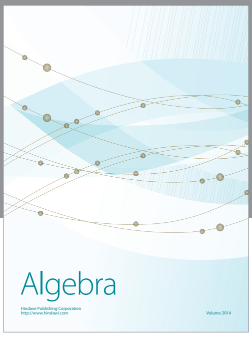
[36] G.-R. Xue, C. Lin, Q. Yang et al., "Scalable collaborative filtering using cluster-based smoothing," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05)*, pp. 114–121, Salvador, Brazil, August 2005.

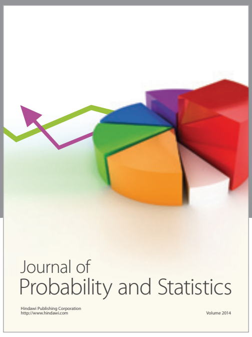
[37] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, Article ID 421425, 19 pages, 2009.


Advances in
Operations Research

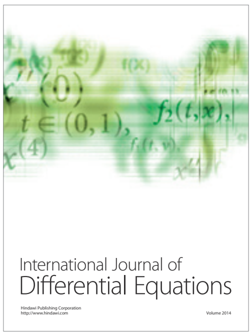

Advances in
Decision Sciences


Journal of
Applied Mathematics

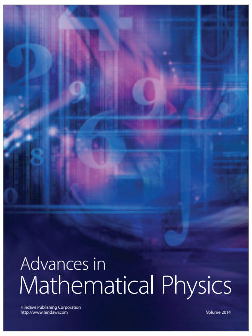

Algebra

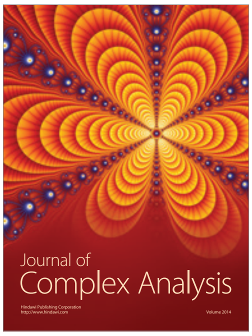

Journal of
Probability and Statistics

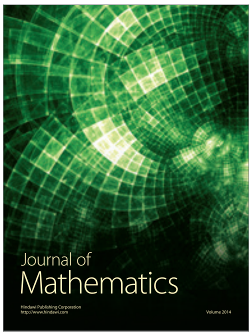
The Scientific
World Journal

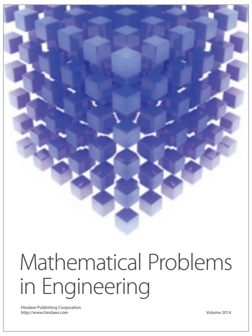

International Journal of
Differential Equations

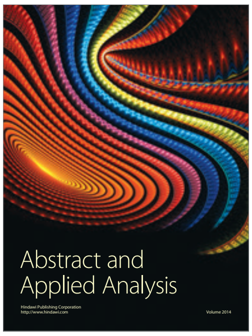
International Journal of
Combinatorics

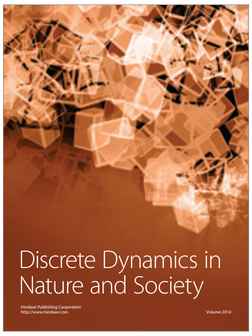

Advances in
Mathematical Physics

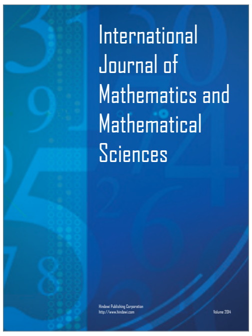

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

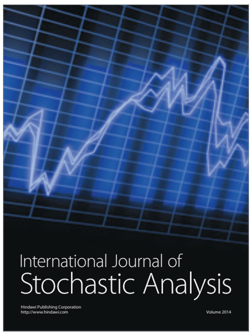

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization